Bleeding segmentation of surgical images

Tian Zhao
Technical University Dresden
Faculty of Computer Science
Dresden, Germany

NCT
NATIONAL CENTER
FOR TUMOR DISEASES
PARTNER SITE DRESDEN
UNIVERSITY CANCER CENTER UCC

Supported by:
German Cancer Research Center
University Hospital Carl Gustav Carus Dresden
Carl Gustav Carus Faculty of Medicine, TU Dresden
Helmholtz-Zentrum Dresden-Rossendorf

Project introduction

- Main purpose

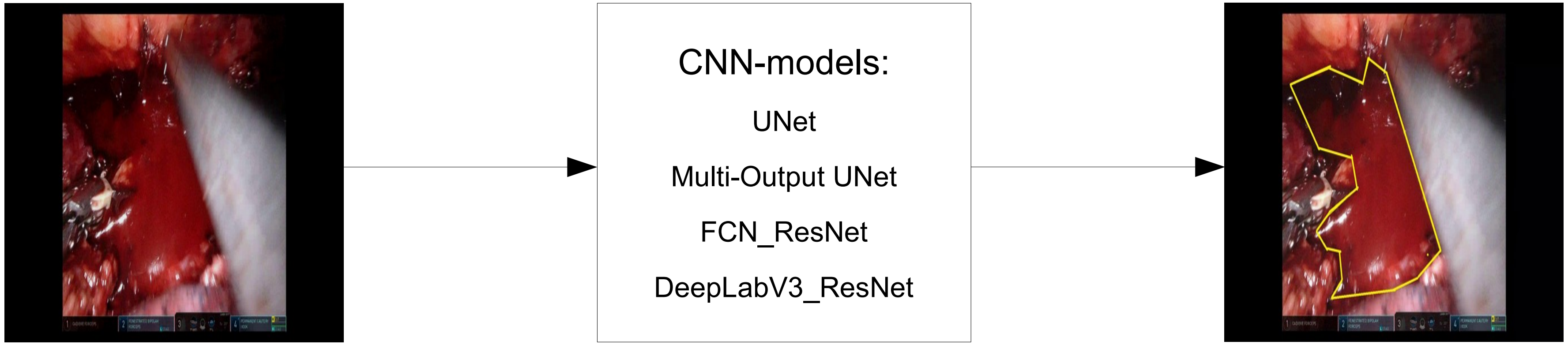


Fig 1. Segmentation process

- Evaluation methods

    dice loss, precision, recall, specificity, f_1

Dataset

- Origional images

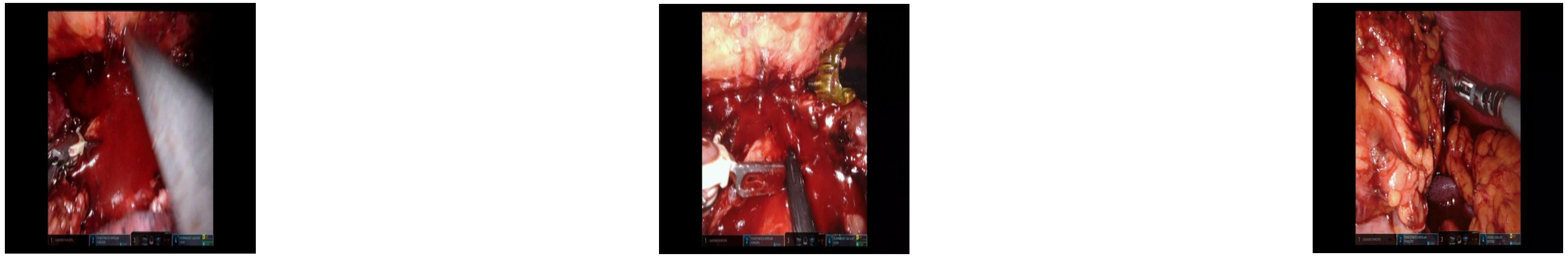


Fig 2. original surgical images

- Polygon annotations

In form of: $[(x_0,y_0), (x_1,y_1), \ldots (x_n,y_n)]$

- Generated masks

Fig 3. generated masks

Dataset

- Size
  - Number of surgeries: 9
  - Number of images **WITH** blood: 946 (70%)
  - Number of images **WITHOUT** blood: 405 (30%)

- K-Ford validation (k=5) grouping rules
  - Absolutely separated training and evaluation sets
  - Minimum difference between size of groups

- Validation group size

  [250, 250, 250, 250, 250]

Pre-Process

- Random-crop and flip (all models)

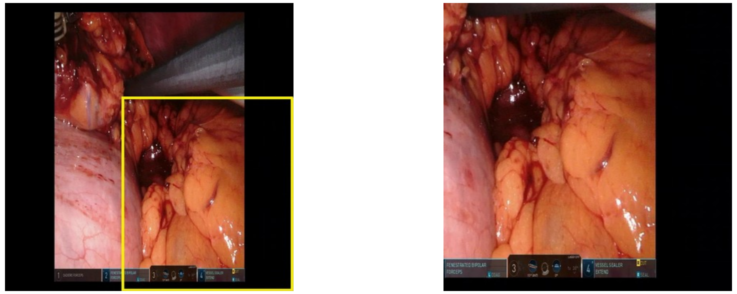


Fig 4. random crop

RandomResizedCrop((256, 256), scale=(0.5, 1.0))

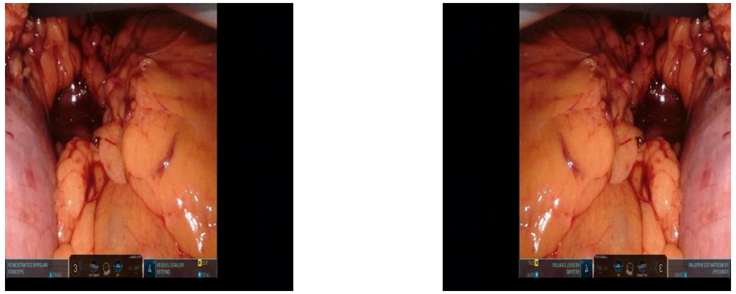


Fig 5. random flip

RandomHorizontalFlip()

- Normalisation (ResNet only)

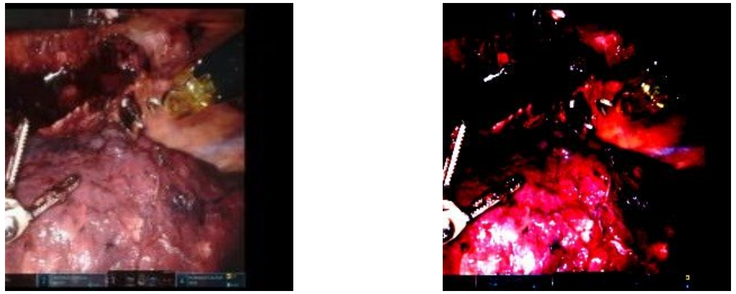


Fig 6. normalization

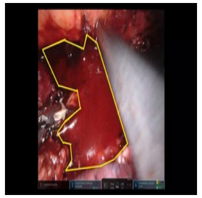
```
Normalize(   mean  = [0.485, 0.456, 0.406],
             std   = [0.229, 0.224, 0.225]     )
```

Evaluation methods

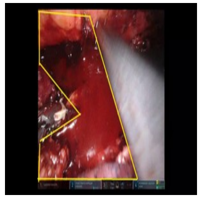
- Dice loss

$$DiceLoss(I_m, I_e) = 1 - \frac{2 \times TP(I_m, I_e) + s}{Positiv(I_m) + Positiv(I_e) + s}$$

- $TP(I_m, I_e) = \sum_{x, y \in (0, 255)} I_m(x, y) \times I_e(x, y)$
- $Positiv(I) = \sum_{x, y \in (0, 255)} I(x, y)$

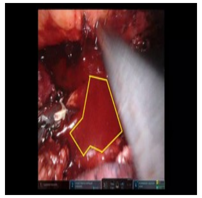


ground truth



too large

$Positiv(I_e)\ \uparrow$ $DiceLoss(I_m, I_e) \uparrow$



too small

$Positiv(I_e)\ \downarrow$
$TP(I_m, I_e)\ \downarrow\downarrow$ $DiceLoss(I_m, I_e) \uparrow$

Fig 7. estimation group

Evaluation methods

- Confusion matrix

		Real category	
		Positive	Negative
Estimation	Positive	**TP**	**FP**
	Negative	**FN**	**TN**

- Measurments

$$\boldsymbol{precision} = \frac{TP}{TP+FP}$$

$$specificity = \frac{TN}{TN+FP}$$

$$recall = \frac{TP}{TP+FN}$$

$$f_1 = \frac{2 \times precision \times recall}{precision + recall}$$

NCT

Models

- Convolution unit

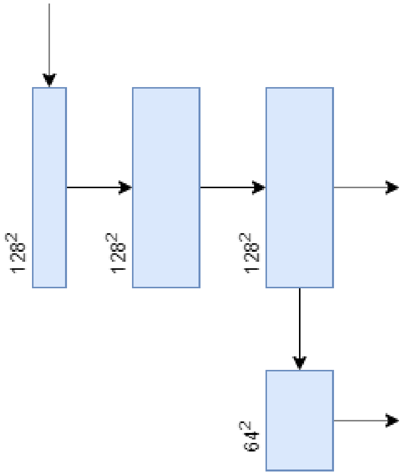


Fig 8. down sampling unit

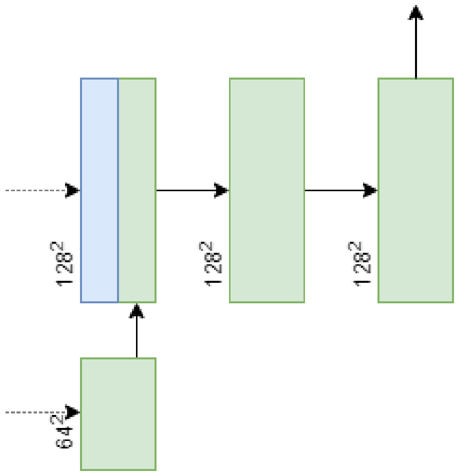


Fig 9. down sampling unit

Models

- UNet

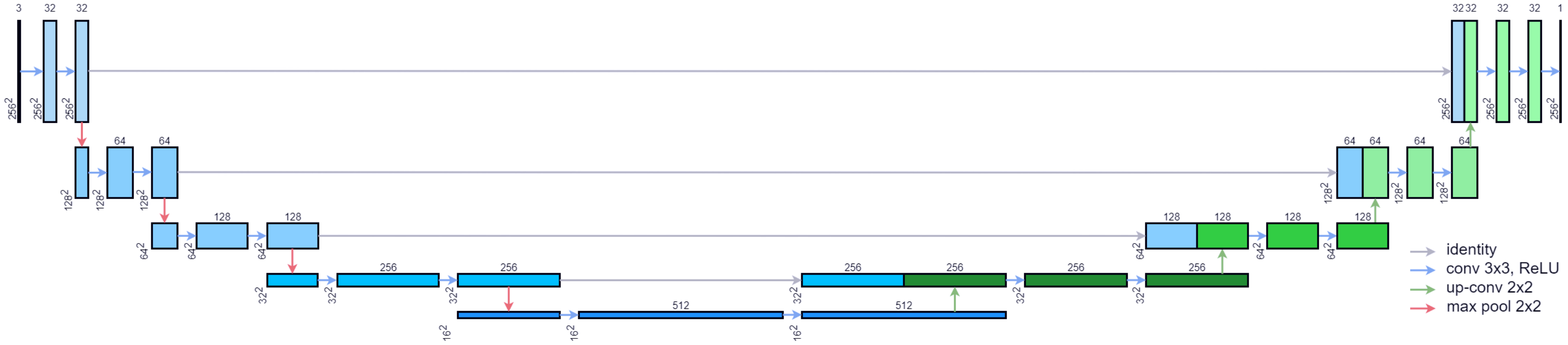


Fig 10. UNet model structure.

Adapted from: [1] Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation.

$$Loss(I_m, I_e) = DiceLoss(I_m, I_e)$$

Models

- Multi-Output UNet

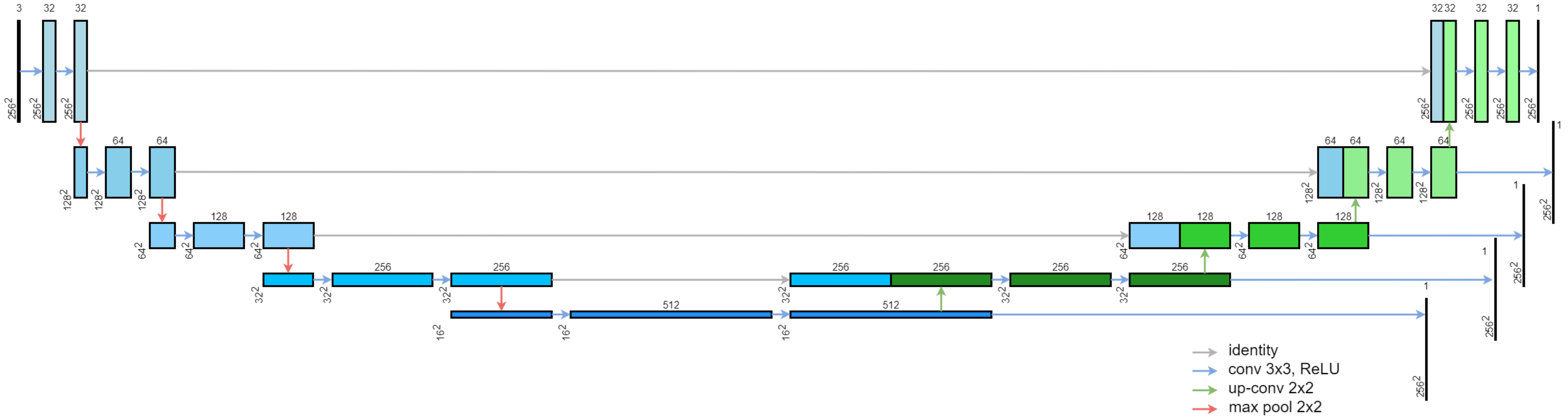


Fig 11. Multi-OutputUNet model structure.

Idea from: [2] Sun, Tao, et al. ”Stacked U-Nets With Multi-Output for Road Extraction.” CVPR Workshops. 2018.

$$Loss(I_m, I_i \in [0,4]) = \frac{\sum_{i \in [0,4]} DiceLoss(I_m, I_i)}{5}$$

Models

- Learning rate

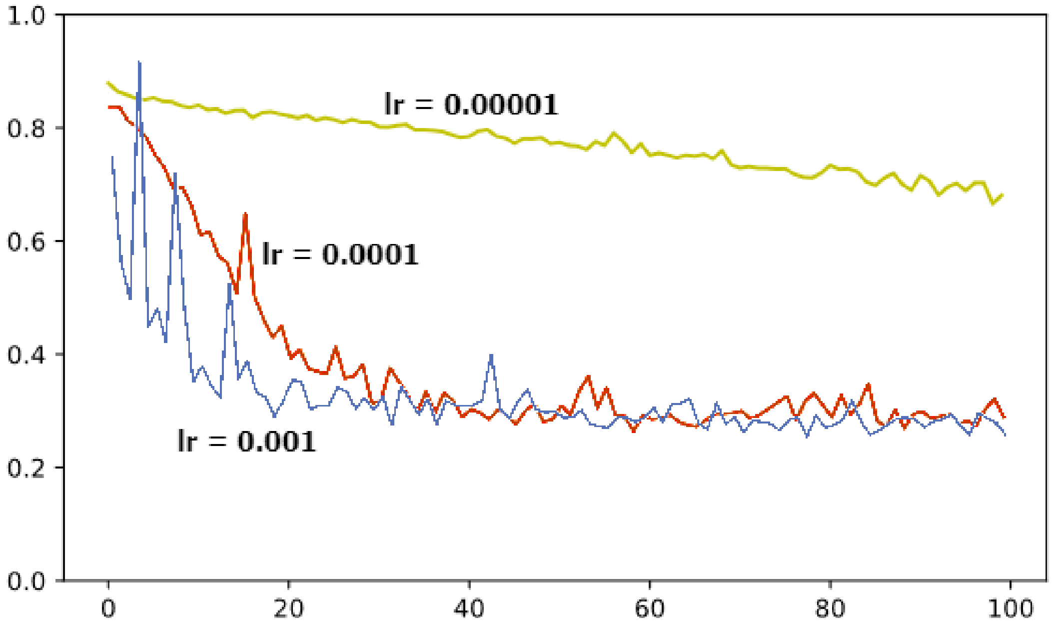


Fig 12. Curve of loss with different learning rate

Results

- Training curve

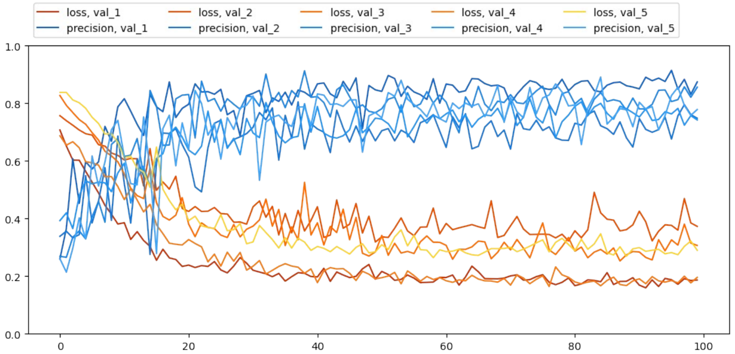


Fig. 13: Evaluation curve of UNet

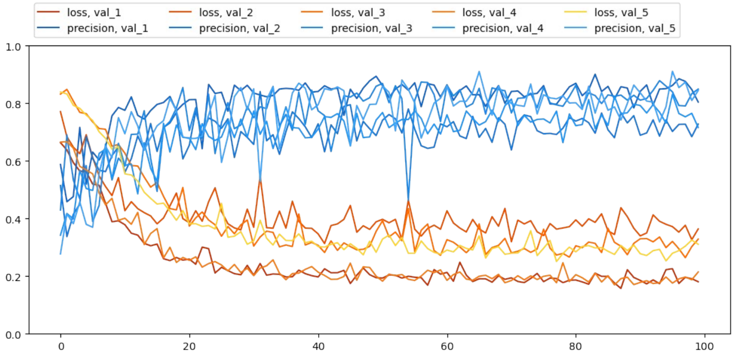


Fig. 14: Evaluation curve of Multi-Output UNet

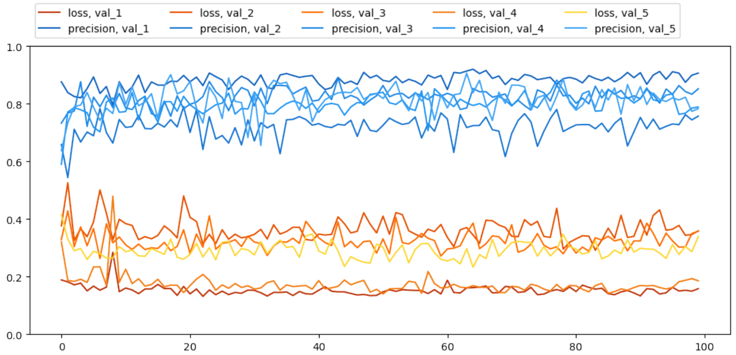


Fig. 15: Evaluation curve of FCN ResNet50

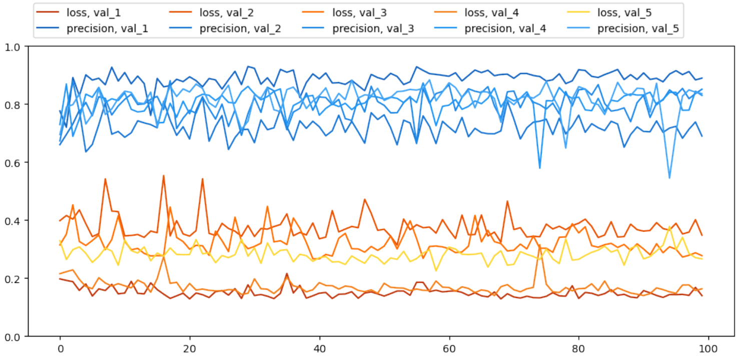


Fig. 16: Evaluation curve of DeepLabV3 ResNet50

Results

- Model estimations

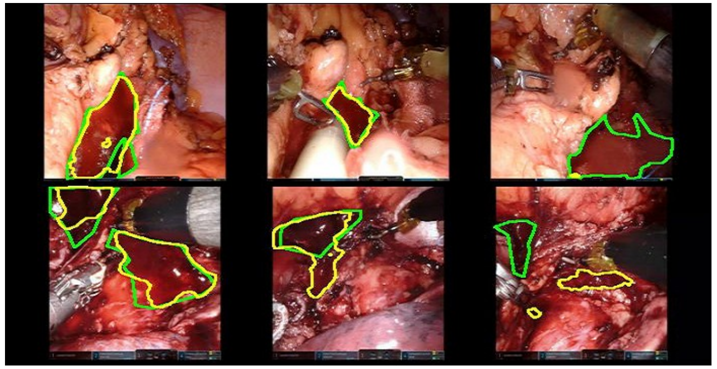


Fig. 17: Evaluation curve of UNet

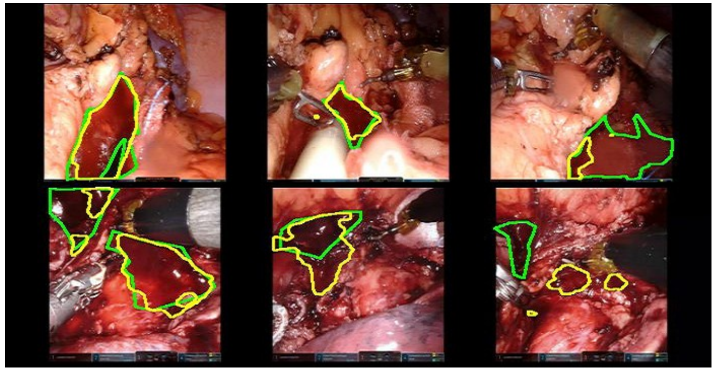


Fig. 18: Evaluation curve of Multi-Output UNet

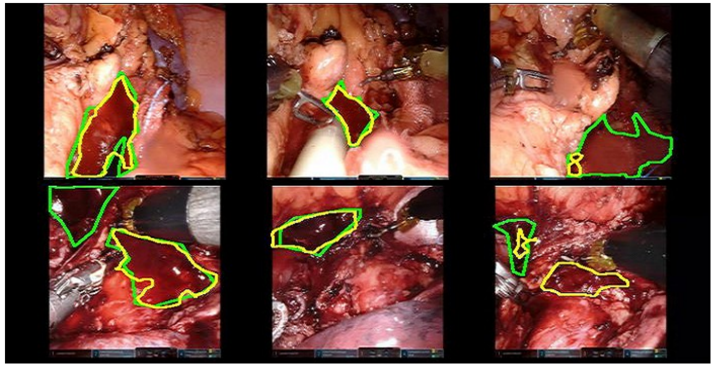


Fig. 19: Evaluation curve of FCN ResNet50

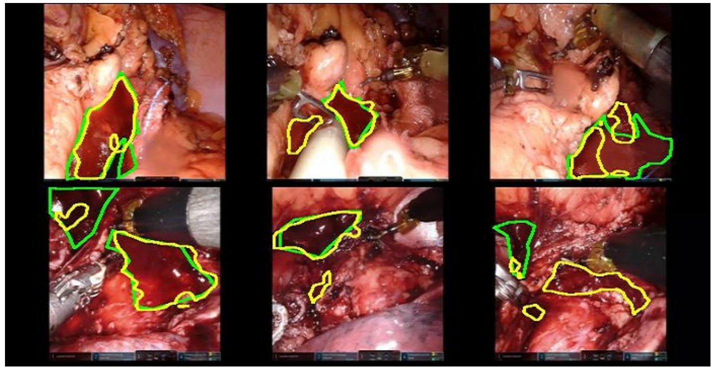


Fig. 20: Evaluation curve of DeepLabV3 ResNet50

Results

- Evaluation data and statistic

	loss	precision	recall	specificity	f_1
cross_val 1	0.1865	0.8739	0.7618	0.9896	0.8135
cross_val 2	0.3727	0.7479	0.5571	0.9848	0.6273
cross_val 3	0.3065	0.8558	0.5912	0.9956	0.6935
cross_val 4	0.1958	0.7421	0.8942	0.9672	0.8042
cross_val 5	0.2897	0.7781	0.6560	0.9918	0.7103
average	0.2702	0.7996	0.6920	0.9858	0.7298

Fig. 21: Evaluation curve of UNet

	loss	precision	recall	specificity	f_1
cross_val 1	0.1806	0.8042	0.8409	0.9782	0.8194
cross_val 2	0.3637	0.7276	0.5727	0.9824	0.6363
cross_val 3	0.3284	0.8490	0.5691	0.9954	0.6716
cross_val 4	0.2144	0.7159	0.8763	0.9644	0.7856
cross_val 5	0.3107	0.8441	0.5859	0.9955	0.6892
average	0.2796	0.7882	0.6890	0.9832	0.7204

Fig. 22: Evaluation curve of Multi-Output UNet

	loss	precision	recall	specificity	f_1
cross_val 1	0.1582	0.9062	0.7875	0.9917	0.8418
cross_val 2	0.3586	0.7570	0.5697	0.9850	0.6414
cross_val 3	0.3576	0.8517	0.5684	0.9954	0.6424
cross_val 4	0.1856	0.7889	0.8818	0.9745	0.8144
cross_val 5	0.3386	0.7830	0.6201	0.9925	0.6614
average	0.2797	0.8173	0.6855	0.9878	0.7203

Fig. 23: Evaluation curve of FCN ResNet50

	loss	precision	recall	specificity	f_1
cross_val 1	0.1395	0.8891	0.8406	0.9892	0.8605
cross_val 2	0.3478	0.6894	0.6597	0.9754	0.6522
cross_val 3	0.2772	0.8508	0.6857	0.9942	0.7228
cross_val 4	0.1630	0.8296	0.8807	0.9811	0.8370
cross_val 5	0.2662	0.8358	0.6541	0.9939	0.7338
average	0.2387	0.8189	0.7442	0.9868	0.7613

Fig. 24: Evaluation curve of DeepLabV3 ResNet50

NCT

Results

- Response time

	CPU (i7-6700k)	GPU (GeForce RTX 2070)
UNet	110ms (9.09 fps)	2.5ms (400 fps)
Multi-Output UNet	110ms (9.09 fps)	2.8ms (357 fps)
FCN_ResNet50	270ms (3.70 fps)	5.8ms (172 fps)
DeepLabV3_ResNet50	440ms (2.27 fps)	6.2ms (161 fps)

Fig. 25: Response time of different model

Conclution

	Loss	Stability	Response Time
UNet	medium	medium-low	short
Multi-Out UNet	medium	medium	short
FCN ResNet	medium	medium	long
DeepLab ResNet	medium	medium	long

Reference

[1] Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation.

[2] Sun, Tao, et al. ”Stacked U-Nets With Multi-Output for Road Extraction.” CVPR Workshops. 2018.